Remarks and Arguments

Claims 1-37 were presented for examination. Claims 17-37 were canceled as drawn to non-elected inventions and these claims will be the subject of further divisional applications. Claims 1, 7, 8, 12, 14 and 15 have been amended. Claim 16 has been canceled.

The specification has been amended at page 2 in order to insert available identifying information for various references to other applications.

Claim 7 has been rejected under 35 U.S.C. §112, second paragraph, for insufficient antecedent basis for the term "dependency identifier." Claim 7 has been amended to recite, in lines 1-3, "...the data change request priority scheme includes a dependency identifier in each data change request..." Thus, the recitation of the "dependency identifier" in claim 7, line 4 finds antecedent basis in claim 7, line 3.

The examiner has indicated that, if claim 8 is found allowable, claim 9 will be objected to as a substantial duplicate of claim 8. In response, claim 8 has been amended to recite, in lines 1-2, that the dependency identifier specifies at least one data change request. Claim 9 recites in lines 1-2 that the dependency identifier specifies one data change request. Therefore, claim 9 clearly has a different scope from amended claim 8.

Claim 1 has been rejected under 35 U.S.C. §102(a) as anticipated by an article entitled, "Amaze: A Multiplayer Computer Game" by Eric C. Berglund and David Cheriton (Berglund.)

The present invention is addressed to a peer-to-peer collaboration system in which collaborators maintain local copies of the same collaborative data and each collaborator can make changes to the collaborative data. When a collaborator makes a change to his local data copy, this change is then sent directly to all other collaborators with a data change request or delta.

In such systems, a problem occurs in maintaining consistency among the local data copies. In particular, the transit time for the data change requests can vary between collaborators, especially if a public network, such as the Internet, is used to connect the collaborators. In many cases, the order in which the data changes are made is critical and, consequently, some mechanism must be used to insure that the data changes are applied in the same order to all local data copies.

In accordance with the principles of the invention, in each collaborator, a dynamics manager implements a data change request priority scheme that determines an order for executing the data change requests to promote data consistency. In one embodiment of the data change request priority scheme, each data change request has associated therewith, and preferably contains, order information and dependency information. The dynamics manager can use this information to determine an order in which to apply the requested data changes to change the local data copy.

However, in some cases, due to the aforementioned latency problem, a data change request from a remote collaborator will arrive at a local collaborator and the dynamics manager will determine (from the order and dependency information in the new request) that this newly-received data change request should have been made before other already-made data change requests. In this case, the dynamics manager can selectively cause a roll-back of data changes that have already been made by "undoing" the effects of these changes until the local data copy is in a state at which the newly received data change request should have been made. The dynamics manager can then cause the newly received data change to be made and may then remake all of the subsequent data changes thereby resulting in a new order of data changes that reflects the newly received change.

The Berglund reference discloses a distributed game system that recognizes the data copy consistency problem as set forth at page 34, column, 1, second paragraph, but chooses to solve this problem by "partitioning" the game state so that each player can update only the game state associated with their character. With this partitioning, there can be no update conflicts regardless of the order of arrival of update messages. This is clearly stated in Berglund at page 34, column 1. It is also clear from Berglund Figure 2, which shows that the first monster's state is always updated by player 1 and that the second monster's state is always updated by player 2. Since there can be no conflicts, the roll-back and remaking operations performed by the inventive dynamics manager is unnecessary and consequently not disclosed or suggested by Berglund.

Claim 1 has been amended to more particularly point out and distinctly claim the dynamic manager. For example, amended claim 1, in lines 16-21, now recites a dynamics manager, "...wherein the dynamics manager, responsive to the data change requests, can cause the making of selected data changes in an order, the rolling-back of the selected data changes and the remaking of the selected data changes in another order so that the local copy of the data is consistent with a copy of the data maintained by the remote network-capable device." Berglund discloses nothing regarding the rolling back and remaking of data changes in order to change the order of the changes, since in Berglund, the order is immaterial. The Berglund "helper" processes that the examiner analogizes to the inventive dynamics manager, are required in Berglund to read and report any state updates made to game character states by remote players who "own" the characters and are the only ones that can update the states. It is clear that the Berglund helper processes do not modify the game data and specifically do not alter the order of the state changes as recited in claim 1. Thus, amended claim 1 patentably distinguishes over the cited Berglund reference.

Claims 2-16 have been rejected under 35 U.S.C. §103(a) as obvious over the Berglund reference in view of U.S. Patent No. 5,802,322 (Niblett.) The examiner comments that Berglund discloses most of the claimed material except that it does not use sequence numbers as recited in claims 2, 3, 12, 14 and 15. However, the examiner claims that Niblett shows such sequence numbers and that it would have been obvious to combine Berglund and Niblett in order to more completely serialize updates.

The Niblett patent discloses a token passing conference ring network in which a "permit-token" is used to serialize updates. The permit-token is passed from node to node and contains an update number. When a node has possession of the token it can perform an update and increment the update number. Each node maintains a queue of updates received and the current update level of the updates that have been applied to the local data set. When updates are performed, the queue is searched so that the updates are performed in order. If an update at a particular level is missing, the node waits until the update has been received before applying later updates. This operation is clearly set forth in the Niblett abstract.

Although the examiner suggests that it would be proper to combine Berglund and Niblett because they are analogous art, there is no motivation to combine these references because, as stated above, Berglund is designed so that the update order is irrelevant. Thus, there is no reason to introduce the Niblett serialization arrangement because, in Berglund, serialization is not required. Further, even if these references are combined, they still do not teach or suggest the claimed invention. In particular, the serialization arrangement of Niblett is designed to guarantee that all updates will be made in the same order at each node because each node waits until it has the required updates and then applies them based on the update level. This is clearly described in Niblett, column 7, line 65 to column 8, line 14. This effectively insures that all nodes apply all updates in the same order. Thus, combining Berglund and Niblett could suggest a system in which all state updates are serialized so that all updates are applied in the same order at all Berglund player locations. However, it should be noted that the Niblett system only operates in a token ring network that allows serialization by means of the permit-token.

The combination of the Berglund and Niblett references does not suggest the dynamics manager rollback and remaking operations that are recited in amended claim 1 (on which claims 2 and 3 depend) as discussed above. This is because the Niblett serialization mechanism guarantees that an update cannot arrive "out-of-order" after other updates have been applied as in the present invention. Therefore, there is no reason to undo updates and then redo them in a different order as recited in amended claim 1 lines 16-21 as discussed above. Consequently, amended claim 1 patentably distinguishes over the cited combination of Berglund and Niblett.

Claims 2-11 are dependent, either directly or indirectly on amended claim 1 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination in the same manner as amended claim 1. In addition, these claims recite additional limitations not shown or suggested by the cited combination. For example, claims 2, 7 and 11 recite the roll back and remaking operations recited in claim 1 and not taught or suggested by the cited combination.

Claims 12, 14 and 15 have been amended in a similar manner to claim 1 and distinguish over the cited combination of Berglund and Niblett in the same manner as amended claim 1. For example, claims 12 recites in lines 10-18 a dynamics manager "...that receives locally-generated data change requests and remotely-generated data change requests for determining, responsive to information contained in the locally-generated and remotely-generated data change requests, an order in which requested changes are made to the local copy of the data, for making requested data changes in the determined order, for rolling-back data changes made when a remotely-generated data change request is received out of the determined order and for remaking requested data changes in a new order." As discussed above, the rollback and remaking operations are not taught or suggested by the cited combination of references. Consequently, amended claim 12 patentably distinguishes over the cited references and their combination.

Claim 13 is dependent on amended claim 12 and incorporates the limitations thereof. Therefore, it distinguishes over the cited combination in the same manner as amended claim 12.

Claim 14, in lines 8-13, recites a "...dynamics manager responsive to dependency information contained in the deltas for determining an order for processing the deltas to make changes to each local data copy, the dynamics manager being responsive to the reception of a remotely-generated delta out of the determined order for causing the selective rollback of changes made to a local data copy and selective remaking of the rolled-back data changes in a new order." Similarly, claim 15, in lines 10-18, recites the steps of (B) determining an order for processing the deltas based on sequence information contained within the deltas, (C) processing the deltas in the determined order thereby making changes to a local data copy as requested by the deltas, (D) rolling back requested changes made to the local data copy in response to sequence information contained within the deltas indicating that a remotely-generated delta has been received out of the determined order, and (E) remaking the rolled-back data changes in a new order. As discussed above, the rollback and remaking operations are not taught or suggested by the cited combination of references.

Consequently, amended claims 14 and 15 patentably distinguish over the cited references and their combination.

In light of the forging amendments and remarks, this application is now believed in condition for allowance and a notice of allowance is earnestly solicited. If the examiner has any further questions regarding this amendment, he is invited to call applicants' attorney at the number listed below. The examiner is hereby authorized to charge any fees or direct any payment under 37 C.F.R. §§1.17, 1.16 to Deposit Account number 02-3038.

Respectfully submitted

Paul E Kudirka Date: 1/6/03

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656

Version Marked to Show Changes

Page 2, line 9:

Serial No. [XX/XXX,XXX] 09/356,930, entitled "Method and Apparatus for Activity-Based Collaboration by a Computer System Equipped with a Dynamics Manager," filed on even date herewith by Raymond E. Ozzie and Jack E. Ozzie (Atty Docket No. G0008/7001), now U.S. Patent No. 6,446,113 B1.

Page 2, line 13:

Serial No. [XX/XXX,XXX] 09/357,007, entitled "Method and Apparatus for Activity-Based Collaboration by a Computer System Equipped with a Communications Manager", filed on even date herewith by Raymond E. Ozzie, et al. (Atty Docket No. G0008/7000).

1. (Amended) A local network-capable device adapted for collaborative operation and communication over a network with at least one remote network-capable device to enable the local network-capable device and the remote network-capable device to cooperatively edit the same data, said local network-capable device comprising:

A) a memory for storing a local copy of the data in accordance with a data model;

B) a data-change engine coupled with the memory, and responsive to a plurality of data change requests, for controlling storage of the local copy of data in the memory in accordance with the data model and making changes to the local copy of the data; the data change requests including a locally-generated data change request and a remotely-generated data change request; and

C) a dynamics manager, coupled with the data-change engine, and responsive to the data change requests for controlling the engine and coordinating execution of the data change requests; wherein the dynamics manager, responsive to the data change requests, can cause the making of selected data changes in an order, the rolling-back of the selected data changes and the remaking of the selected data changes in another order so that the local copy of the data is consistent with a copy of the data maintained by the remote network-capable device.

7. (Amended) The local network-capable device in accordance with claim 3, wherein the data change request priority scheme includes [encoding the data change requests with] a dependency identifier in each data change request, and the dynamics manager is responsive to the dependency identifier in causing rolling-back and remaking of data changes.

8. (Amended) The local network-capable device in accordance with claim 7, wherein the dependency identifier specifies [a] at least one data change request on which the encoded data change request depends.

12. (Amended) A distributed, coordinated system for maintaining plural copies of the same data pursuant to a distributed data model, wherein the copies can be changed responsive to users' actions, the system comprising:

A) a plurality of computer systems, each of the computer systems capable of locally generating a plurality of data change requests for changing a local copy of the data and of executing data change requests including the locally-generated data change requests and remotely-generated data change requests generated by others of the computer systems so as to make the requested changes to the local copy of the data;

B) each of the computer systems including a dynamics manager that receives locally-generated data change requests and remotely-generated data change requests for determining, responsive to information contained in the locally-generated and remotely-generated data change requests, an order in which [the] requested changes are made to the local copy of the data, for making requested data changes in the determined order, for rolling-back data changes made when a remotely-generated data change request is received out of the determined order and for remaking requested data changes in a new order.

14. (Amended) A framework apparatus for providing communication services for an activity-based collaboration system in which data change requests comprising deltas are communicated over a network between network-capable devices in order to maintain consistency between local data copies, the framework apparatus comprising a communications manager operable on a local network capable device for sending locally-generated deltas over a network to at least one remote network-capable devices and for receiving remotely-generated deltas from the at least one remote network-capable device; and a dynamics manager responsive to dependency information contained in the deltas for determining an order for processing the deltas to make changes to each local data copy, the dynamics manager being responsive to the reception of a remotely-generated delta out of the determined order for causing the selective rollback of changes made to a local data copy and selective remaking of the rolled-back data changes in a new order.

15. (Amended) A method for providing communication services for an activity-based collaboration system, in which data change requests comprising deltas are communicated over a network between network-capable devices in order to maintain consistency between local data copies, the method comprising the steps of:

A) sending locally-generated deltas from a local network-capable device over a network to at least one remote network-capable devices and for receiving remotely-generated deltas from the at least one remote network-capable device;

B) determining an order for processing the deltas based on sequence information contained within the deltas; [and]

C) processing the deltas in the determined order thereby making changes to a local data copy as requested by the deltas;

D) rolling back requested changes made to the local data copy in response to sequence information contained within the deltas indicating that a remotely-generated delta has been received out of the determined order; and

E) remaking the rolled-back data changes in a new order.